

Antti Huovinen

# Verkkokurssin alustaratkaisu ja toteutus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

23.5.2013

Tekijä Otsikko	Antti Huovinen Verkkokurssin alustaratkaisu ja toteutus
Sivumäärä Aika	34 sivua 23.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaajat	muutosvalmentaja Marja-Leena Savimäki yliopettaja Harri Airaksinen
<p>Insinöörityössä oli tavoitteena etsiä verkkokurssille sopiva alustaratkaisu ja suunnitella kurssille sopiva verkkosivu alusta asti ja toteuttaa sen ohjelmointi valitulle alustalle. Työn tarkoituksena oli vapauttaa insinöörityön asiakkaan aikaa siirtämällä koulutuskurssi verkkoon.</p> <p>Insinöörityössä tutustuttiin sisällönhallintajärjestelmiin ja niiden ominaisuuksien hyödyntämiseen verkkosivujen rakentamisessa, valittiin asiakkaalle sopivin sisällönhallintajärjestelmä, suunniteltiin verkkosivun graafinen ulkoasu sekä tehostettiin toimintaa käyttämällä PHP, HTML, CSS ja JavaScript -ohjelmointikieliä. Työssä pohdittiin myös tietoturvaa ja sen vaikutusta käytettävyyteen. Työn laajimmassa osassa tehtiin pohja verkkokurssisivujen esittämiseksi maksaneille asiakkaille ja suunniteltiin siihen toimiva käyttöliittymä.</p> <p>Lopputuloksena saatiin lähes valmis sisällönhallintajärjestelmässä toimiva dynaaminen verkkosivu, jossa on kaksi osaa, maksaneiden asiakkaiden osa ja julkiosa, ja jota voidaan käytännössä soveltaa minkä tahansa vastaavanlaisen verkkokurssin julkaisemiseen. Lisäksi sisällönhallintajärjestelmien tietoturvasta selvisi paljon tietoa, jota hyödynnettiin työssä.</p> <p>Työstä saadut kokemukset osoittavat valitun sisällönhallintajärjestelmän soveltuvuuden myös työssä tulokseksi saadun kaltaiseen järjestelmään.</p>	
Avainsanat	verkkokurssi, sisällönhallintajärjestelmä

Author Title	Antti Huovinen Platform solution and implementation of an online course
Number of Pages Date	34 pages 23 May 2013
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Marja-Leena Savimäki, Coach Harri Airaksinen, Senior Lecturer
<p>The objective of this Bachelor's thesis was to seek a suitable platform solution for an online course and design a suitable web page for it from scratch and to implement it to the chosen platform. The purpose of this thesis was to create more free time for the customer by moving the course online.</p> <p>This thesis explored content management systems (CMS) and how to make use of their abilities when building a web page, choosing the best CMS for the customer, designing a graphical layout for a web page and using PHP, HTML, CSS and JavaScript -programming languages to enhance the performance. This thesis also discusses web security and its effect on usability. This study focused on making a base for an online course accessed by paying customers only and designing a functioning user interface for it.</p> <p>As a result of this project a dynamic web page functioning within a CMS with two parts was almost fully achieved. The web page was designed for paying customers as well as for public use and it can be applied to similar web courses. In addition, a lot of information about CMS's security was revealed, which was put into use during this project.</p> <p>The results of this thesis show that the chosen CMS can be applied into systems such as the one designed in this study.</p>	
Keywords	online course, CMS

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Verkkokurssin suunnittelu	2
2.1	Kurssin suunnittelu	2
2.2	Toteutuksen suunnittelu ja alustan valinta	3
2.3	WordPress alustana	4
2.4	Käyttöliittymäsuunnittelu	5
2.5	Graafinen suunnittelu	6
3	Voimanainen-julkiosan toteutus	8
3.1	Julkiosan tekninen toteutus	8
3.2	Julkiosa käyttäjän ja ylläpitäjän näkökulmasta	13
4	Voimanainen-kurssiosan toteutus	16
4.1	Kurssiosan tekninen toteutus	19
4.2	Kurssisivusto käyttäjän näkökulmasta	24
4.3	Ylläpitäjän näkökulma	24
5	Sivuston dynaamisuus ja testauksen aloitus	29
6	Yhteenveto ja tulevaisuus	31
	Lähteet	33

## Lyhenteet

CSS	Cascading Style Sheets. Tyylitiedostoissa käytetty ohjelmointikieli.
DOM	Document Object Model. Ohjelmointirajapinta HTML:lle.
HTML	HyperText Markup Language. Verkkosivujen pääohjelmointikieli.
JavaScript	Selaimen puolella suoritettava ohjelmointikieli.
jQuery	Hyvin yleisesti käytetty JavaScript-kirjasto.
LAMP	Linux, Apache-http-palvelin, MySQL ja PHP, Perl tai Python. Avoimen lähdekoodin palvelinratkaisu.
PHP	Hypertext Preprocessor. Palvelimen puolella käytettävä ohjelmointikieli.
WYSIWYG	"What you see is what you get". Tekstimuokkain.

## 1 Johdanto

Insinööriyön tarkoituksena on luoda Marja-Leena Savimäen vetämää Voimanainen-ihmiskoulutuskurssia varten sopiva verkkokurssialusta. Tarkoitus on saada aikaan monimuotoisen tiedon välittämiseksi alusta, jota voidaan mahdollisesti käyttää myös muiden vastaavien kurssien ulkoistamiseen.

Marja-Leena Savimäki on kouluttaja ja muutosvalmentaja, joka on työskennellyt vuodesta 1997 asti alallaan osuuskuntayrittäjänä, kunnes vuonna 2001 perusti toiminimen, jonka alla hän on sittemmin jatkanut työskentelyä. Voimanaisen lisäksi Savimäki on kouluttanut muitakin aiheita ja tehnyt yksilövalmennuksia ja työnohjauksia. Hän on vetänyt Voimanainen-kurssia vuodesta 1998 asti kansalaisopistossa, yrityksissä ja aikuisopistossa, mutta hän on joutunut ajanpuutteen vuoksi harventamaan kurssin vetämistä kahteen kertaan vuodessa. Kurssille on kuitenkin edelleen paljon kysyntää, ja tästä syystä verkkokurssi on oivallinen ratkaisu, sillä se ei vie juurikaan kouluttajan aikaa ja toisaalta se tarjoaa asiakkaalle mahdollisuuden päästä osallistumaan Voimanainen-kurssille myös Suomen ulkopuolelta.

Haasteelliseksi insinööriyön tekee se, että valmista alustaa juuri tälle kurssille ei välttämättä löydy, ja jos löytyykin, se ei todennäköisesti ole budjettiin sopiva. On siis todennäköistä, että verkkokurssin alusta rakennetaan jonkin ennalta hyväksi todetun ilmaisen järjestelmän pohjalle, jota voidaan muokata verkkokurssin tarpeisiin sopivaksi. Kurssin vetäjän, Savimäen, tarpeina kouluttajan roolissa ovat tekstimuotoisen tiedon jäsentely halutulla tavalla, videoiden esittäminen, kurssilaisen aktivointi tehtävien muodossa, hallinnan helppous ja verkkomaksuominaisuus.

Voimanainen-verkkokurssin on tarkoitus rakentua siten, että käyttäjälle esitetään Savimäen kirjoittamia tekstejä ja videoita ja niihin liittyen esitetään kysymyksiä, joiden tarkoitus on herättää ajatuksia ja auttaa sekä ohjata muutosprosessissa. Insinööriyön tarkoituksena ei ole suunnitella ja rakentaa perinteistä verkko-oppimisympäristöä, jollainen usein on rakennettu esimerkiksi Moodle-järjestelmään, vaan tarkoitus on kehittää ratkaisumalli verkossa myytävän ja sinne rakennettavan kurssin toteuttamiselle.

## 2 Verkkokurssin suunnittelu

### 2.1 Kurssin suunnittelu

Suunnittelun alussa piti miettiä, mitä oikein oltiin tekemässä, sillä tavoitteena ei ollut perinteinen verkko-oppimisympäristö. Voimanaisten tapauksessa ei ollut tarkoitus tehdä tiukasti hallittua ympäristöä, jossa jokaisen kurssiviikon suorituksella ja etenemisellä olisi jokin vaade. Tarkoitus oli alusta asti luoda multimedialle alusta, jota voisi käyttää itsensä kouluttamiseen maksua vastaan. Koulutus perustuisi ainoastaan asiakkaan omaan haluun, minkä takia olisikin erittäin tärkeää, että käyttöliittymä ja käyttöympäristö olisivat mahdollisimman miellyttävät.

Opinnäytetyön asiakkaalla oli yksi selkeä kiinnostus oman tulevan asiakaskuntansa puolesta: tietoturva. Huoli tietoturvasta oli suurehko siitä syystä, että kurssilaisille annetaan kurssin aikana tehtäviä, joiden vastaukset ovat hyvin henkilökohtaisia ja mahdollisesti arkaluontoisia. Tietoturva olikin yksi projektin kulmakiviä.

Mitään pohjaa verkkokurssille ei ollut, joten projektin aloituksessa ainoa käytettävä materiaali oli asiakkaan käyntikortti, jossa oli kurssilogo ja värimaailma, jotka asiakas halusi säilyttää ennallaan ja joiden tuli siten näkyä myös sivuston ulkoasussa. Tätä toivotta kunnioittaen sivustolle alettiin suunnitella Voimanaisten tyyliin sopivaa tunnelmallista ulkoasua, jossa tarkoituksena oli tuoda esille logon zeniläistä tyyliä.

Suunnittelun alkuvaiheilla näytti siltä, että asiakkaan tuotteen myynti ja toimitus tulisi toimimaan erillisen laskituksen kautta, jota olisi seurannut manuaalinen käyttöjäoikeuksien korotus maksaneelle asiakkaalle. Jälkikäteen tästä kuitenkin luovuttiin, sillä siitä pohdittiin tulevan liian paljon teknisiä ja ajankäyttöön liittyviä ongelmia, etenkin kun yksi verkkokurssin suurimpia tavoitteita oli nimenomaan saada se kuluttamaan mahdollisimman vähän Savimäen aikaa muiden kiireiden takia.

Kurssin suunnittelussa asiakas oli vahvasti mukana. Hän esitti ajatuksia siitä, mitä pitäisi ottaa huomioon suunnittelussa ja toteutuksessa. Osa ajatuksista oli teknisesti toteutettavia, osa ei, ja osa oli toteutettavia, mutta epätodennäköisesti toteutukseen mahdollisia. Suunnittelu alkoi pääasiassa siitä ajatuksesta, että kurssin tulisi toimia siten, että sillä olisi neljä kuukautta suoritusaikaa. Asiakas oli suunnitellut kurssin rakennetta si-

ten, että se jaettaisiin kymmeneen suorituskertaan, jotka voisi suorittaa kymmenenä viikkona. Tämä jättäisi käyttäjälle neljä väliviikkoa. Suoritus suunniteltiin aluksi siten, että käyttäjä ei voisi enää palata takaisin aikaisemmin suorittamalleen viikolle, mutta tämä todettiin jo silloin mahdottomaksi toteuttaa, sillä sivut jäisivät kuitenkin selaimen välimuistiin.

Kurssisuoritus pitäisi pystyä jotenkin tallentamaan, jotta käyttäjän ei tarvitsisi itse pitää kirjaa siitä, mihin kohtaan jäi viimeksi. Aluksi mietittiin myös jokaisen tehtävän vastauksen tarkistusta sen suhteen, että vastaukseksi pitäisi kirjoittaa jotain päästäkseen eteenpäin. Tästä kuitenkin luovuttiin jo suunnitteluvaiheessa, sillä sitä pidettiin heikkona ratkaisuna. Käyttäjä olisi voinut kirjoittaa mitä tahansa, ja suorituksesta olisi tullut epämiellekästä.

Kurssille suunniteltiin myös eräänlainen muistivihko, jota jokainen käyttäjä voisi käyttää muistiinpanoille. Sen tarkoitus olisi toimia kurssin sivussa, ja se tallennettaisiin jokaisen istunnon lopussa. Lisäksi kurssin oheismateriaaliksi suunniteltiin DVD:tä ja Voimanainen-kalenteria. Näiden sisällytystä ensimmäiseen versioon pidettiin epätodennäköisenä, mutta niiden lisääminen jälkikäteen pitäisi ottaa huomioon toteutuksessa.

## 2.2 Toteutuksen suunnittelu ja alustan valinta

Toteutuksen suunnittelun vaateet luonnollisesti määräävät alustan valinnan, koska alustan on kyettävä noudattamaan niitä. Vaatimukset verkkokurssin alustalle olivat melko selkeät toteutuksen kannalta: helposti muokattava, mielellään entuudestaan tuttu tekniikka ja edullinen. Helppo muokattavuus ei tarkoita pelkästään ylläpitäjän muokkausominaisuuksia, vaan vastaamiskykyä sivuston rakenteeseen pohjautuviin haasteisiin, mikä karsikin esimerkiksi Moodlen pois sen kankeahkon rakenteen vuoksi. Muutoinkaan sivuston tarkoitus ei ollut missään vaiheessa vaikuttaa tavalliselta sisäverkkotyylisiin rakennetulta oppimisympäristöltä, vaan enemmänkin modernin verkkosuunnittelun, kirjallisen ilmaisun ja multimedian kohtaamispaikalta.

Valitun alustan tuli myös täyttää teknisistä vaatimuksista sellaiset, joita pystyttiin sille jo alussa asettamaan, kuten muun muassa helppo käyttöönotto, käyttäjien hallinta, sisälönhallinta, sisällön suojaus ja teemoitus. Helposti käyttöönotettavana järjestelmänä



pidettiin esimerkiksi LAMP:iin (Linux, Apache, MySQL, PHP) pohjautuvaa järjestelmää, jonka käyttö Metropolian palvelimella olisi ollut helppoa.

Sisällönhallintajärjestelmistä Drupal oli erittäin varteenotettava järjestelmä sen ylivertaisen muokattavuuden ja muutenkin hyvän käytettävyyden vuoksi. Drupal olisi ollut myös erittäin edullinen, koska se on avoimeen lähdekoodiin perustuva järjestelmä. Alustaksi valittiin kuitenkin WordPress, joka täytti kaikki kolme ehtoa: se on ilmainen, helposti muokattava ja kehitysympäristönä työn tekijälle entuudestaan tuttu. [1; 2.]

## 2.3 WordPress alustana

WordPress on avoimen lähdekoodin sisällönhallintajärjestelmä, joka sai alkunsa vuonna 2003. Henkilökohtaisessa blogikäytössä WordPressiä käyttävät miljoonat ihmiset, mikä tekee siitä maailman suurimman blogityövälineen. Koska WordPress on avoimen lähdekoodin järjestelmä, se on myös ilmainen. WordPressiä työstävät sadat käyttäjät ympäri maailmaa. [2.]

Pääasiassa blogikäyttöön tarkoitettu WordPress tarjoaa valmiiksi paljon hyviä työkaluja, joita projektissa hyödynnettiin aina, kun siihen tarjoutui mahdollisuus. Verkkokurssin rakenne suosii WordPressiä, sillä Voimanainen-verkkokurssi koostuu hyvin pitkälti teksteistä, kuvista ja videoista, aivan kuten blogitkin. WordPress tarjoaa myös hyvät työkalut erilaisten dokumenttien ryhmiin lajitteluun, mikä oli projektin kokonaisuuden kannalta suuri hyöty. Vaikka WordPress onkin kuuluisa blogiominaisuuksistaan, on siitä kehittynyt vuosien aikana myös varteenotettava vaihtoehto muille sisällönhallintajärjestelmille. Tästä osoituksena on se, että Packt-teknologiakirjallisuuden julkaisija, joka alun perin keskittyi avoimen lähdekoodin ohjelmistoihin, valitsi sen vuoden 2010 avoimen lähdekoodin parhaaksi sisällönhallintajärjestelmäksi ja se valittiin The Critters -palkintogaalassa vuoden 2011 parhaaksi avoimen lähdekoodiin verkkosovellukseksi. [3; 4.]

WordPress perustuu LAMP-tekniikkaan, minkä takia insinööriyön pääohjelmointikieleinä oli PHP. Vaikka WordPress onkin melko helppokäyttöinen perusbloggaajalle, se vaatii PHP-ohjelmointitaitoa, mikäli haluaa tehdä sivustosta juuri haluamansa näköisen. WordPress käyttää myös teemoja, jotka koostuvat pääosin juuri PHP:sta, sekä

HTML:sta ja CSS:sta. Teemoja on pelkästään WordPressin omilla sivuilla toista tuhatta ja lisäksi tuhansia maksullisia ja maksuttomia saatavilla Internetistä. [5.]

WordPress koostuu yksinkertaisimmillaan indeksisivusta, jossa esitetään kutsuja eri sisältöä tuottaville mallipohjasivuille. WordPress hakee ensimmäisenä indeksisivua, ellei kutsua millekään muulle sivulle esitetä. Indeksisivu onkin useimmiten WordPress-sivuston etusivu, ellei erikseen määrätä. Esimerkki: indeksisivulla kutsutaan mallipohjasivua *loop.php*, ja silloin tulevat etusivulle näkyviin kaikki sivustolle kirjoitetut postsivut, koska *loop.php* on määrätty niitä ensisijaisesti hakemaan. Samaa periaatetta käyttäen voidaan hakea sivulle eri komponentteja, kuten erillistä alatunnistetta tai sivupalkkia. [6.]

## **Tietoturva**

Tietoturva on aina tärkeä ottaa huomioon, mutta etenkin silloin kun käyttäjän odotetaan antavan sivustolle henkilökohtaista tietoa, kuten tässä projektissa. Henkilökohtaisiin tietoihin kuuluu käyttäjän perustietojen lisäksi tässä projektissa myös vastaukset erilaisiin tehtäviin.

WordPress tarjoaa valmiita salausfunktioita joiden käyttö on helppoa. Näillä voidaan helposti salata yksittäisiä merkkijonoja vaivatta ja taas verrata tarpeen mukaan käyttäjän syötteisiin, kuten esimerkiksi salasanan tapauksessa. Eräs yksinkertaisimpia salaustapoja on MD5, jota WordPress käyttikin kunnes siirtyi nykyiseen php-pass-järjestelmään. [7]

Projektin henkilökohtaisten tehtävävastausten salaaminen oli kuitenkin lähtökohtaisesti monimutkaisempaa. Vastaukset voitaisiin salata tai jättää salaamatta kokonaan ja tallentaa käyttäjän omien tietojen mukaan, joista niihin käsiksi pääseminen olisi todella vaikeaa. Toisaalta vastausten tallentamisen aiheellisuutta itsessään oli myös pohdittava, tai sitä, tarvitseeko vastauksia tallentaa tietokantaan.

## **2.4 Käyttöliittymäsuunnittelu**

Käyttöliittymäsuunnittelussa oli alusta asti selkeää, että sivustolle tarvittaisiin kaksi osaa: etusivu kaikille nähtäväksi ja kurssisivu, jonne vain maksanut asiakas pääsisi

käsiksi. Julkiselle puolelle materiaalia oli niukasti, joten sivustosta oli parasta tehdä mahdollisimman yksinkertainen ja helppokäyttöinen. Helppokäyttöisyyttä oli tarkoitus toteuttaa hyvällä navigaatiolla ja käyttämällä käyttöliittymäsuunnittelussa vakiintuneita tapoja, sillä käyttäjän on helpompi käyttää jotain, mihin on jo tottunut entuudestaan. Esimerkkinä vakiintuneesta tavasta käyttöliittymäsuunnittelussa voidaan käyttää seuraavaa: sivustonavigaatio vasemmassa sivupalkissa tai yläpalkissa. [8, s. 35.] Navigoinnin hyvään käytettävyyteen ja sen saavuttamiseen paneudutaan lisää luvussa 4.

Käytettävyyden kannalta lopputyössä tärkeimpinä oli selkeästi kaksi osaa: verkkokauppa ja itse kurssisivut. Verkkokaupan tuli toimia saumattomasti, jotta siitä ei syntyisi asiakkaille hankaluuksia, jotka saattaisivat johtaa ostopäätöksen peruuntumiseen. Verkkokauppa tuli myös suunnitella siten, ettei asiakas saa väärää kuvaa ostamastaan tuotteesta tai ettei myöskään asiakas voi mahdollisesti väittää, ettei olisi nähnyt tuotteen kuvausta ja ostoehjoja. Toinen osa, eli itse Voimanainen-tuote, tuli suunnitella siten, ettei asiakkaalle jää mitään epäselvää kurssin suoritustavasta, ennen kuin sen aloittaa. Lisäksi itse kurssin suorittamisesta oli pyrittävä tekemään mahdollisimman helppoa, jotta käyttäjä voi kohdistaa kaiken keskittymisensä itse sisältöön. Suunnitteluvaihe herätti seuraavia kysymyksiä, joihin toteutuksessa pyrittiin vastaamaan: Miten kurssia käyvä henkilö pystyy jatkamaan kurssia sujuvasti eri istuntojen välillä ja miten kurssin rakenteen saa sopiviksi jaksoiksi?

## 2.5 Graafinen suunnittelu

Sivuston julkiosan graafisen suunnittelun perustana käytettiin Marja-Leena Savimäen käyntikorttia, jonka etuosa näkyy kuvassa 1. Käyntikortin logoa olikin tarkoitus käyttää sivuston julkiosan logona, ja väriteeman lainaaminen kortista oli myös asiakkaan toive.



Kuva 1. Marja-Leena Savimäen käyntikortin etupuoli.

Käyntikortin väriteema on kovin kermanen, mutta logo edustaa enemmänkin Kiinasta kotoisin olevaa ja sittemmin Japaniin periytynyttä ja siellä kypsyntä zeniläistä tyyliä. Tätä zeniläistä tyyliä oli tarkoitus hyödyntää julkiosan suunnittelussa, sillä se sopii hyvin myös Voimanainen-kurssin tyyliin. [9.]

Sivuston toiseen osaan, jota vain maksavat asiakkaat pääsevät käyttämään, eli kurssiin, oli tarkoitus suunnitella ulkoasu julkiosasta poikkeavaksi. Tämä johtui siitä, että siinä missä julkiosa on sisällön puolesta kovin suppea mutta graafisesti rikas, on kurssiosa puolestaan sisältönsä puolesta todella kattava ja graafisen suunnittelun haluttiin kunnioittavan sitä. Käytännössä kurssiosan graafisen suunnittelun oli tarkoitus vain toimia käyttöliittymää selkeyttämässä mahdollisimman hillityllä tavalla, jotta itse sisältö pääsee korostumaan. Kurssiosan suunnittelun alussa ei ollut mitään selkeää mallia tai teemaa siitä, miltä sen tulisi näyttää, vaan inspiraationa käytettiin ainoastaan minimalistisiä verkkosivuja ja muita vastaavankaltaisia verkkokurssisivuja.

Insinöörityön asiakas oli pyytänyt ulkopuolista osapuolta suunnittelemaan itse Voimanainen-kurssille grafiikkaa. Grafiikan oli tarkoitus olla pääosin kuvia, joita käytettäisiin kurssin kuvituksessa. Kuvissa oli tarkoitus esiintyä Voimanainen-hahmo, josta ei vielä insinöörityön suunnitteluvaiheessa ollut selkeää kuvaa. Tämä teki suunnitteluvaiheesta haasteellisen, sillä kuvien väriteema ei ollut tiedossa eikä niiden vaatima tila tai suunniteltu esitystapa.

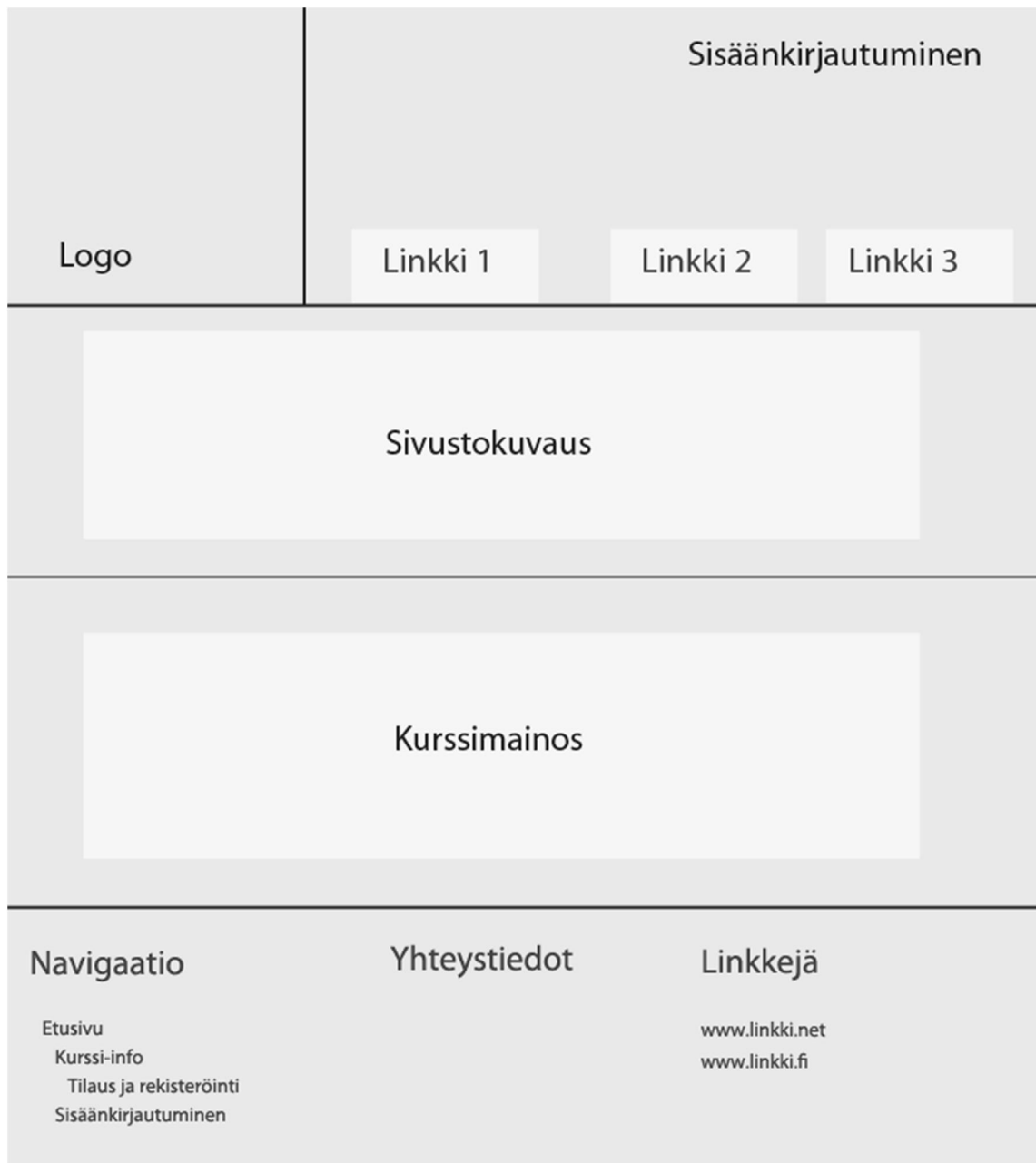
### 3 Voimanainen-julkiosan toteutus

#### 3.1 Julkiosan tekninen toteutus

Voimanainen-verkkosivujen toteutus lähti graafisesta suunnittelusta, joka toteutettiin Adobe Photoshop CS5 -ohjelmaa käyttäen. Suunnittelussa tuli ottaa huomioon, että ulkoasu tulisi pystyä muuntamaan ohjelmoinnilla aikaansaataavaksi samalla säilyttäen hyvä toimivuus. Etusivusta tehtiin useita eri versioita, kunnes asiakas oli siihen tyytyväinen. Etusivun tyyliä noudatettiin myös sivuston alisivuissa.

Etusivu koostuu ylätunnisteesta, pääosan sivustokuvauksesta, kurssimainoksesta sekä alatunnisteesta. Ylätunnisteessa ovat selkeästi Voimanainen-logo ja navigaatio. Ylätunnisteen logosta lähtee koko sivuston halki kulkeva jatko-osa puunrungon muodossa, ja se jatkuu aina alatunnisteeseen asti, jossa se päättyy puun juuriin. Sivuston pääosa jaettiin kahteen osaan: sivustokuvaukseen, joka tässä tapauksessa on Savimäen oma tervehdysteksti, ja kurssimainokseen, jonka taustana hyödynnettiin Savimäen käyntikortin takapuolella olevaa kuvaa. Itse kurssimainos on paperinomaisella pohjalla, jossa teksti juoksee siihen piirtyneillä riveillä. Etusivusta rakennettu kaaviomalli näkyy kuvassa 2.

Alatunniste koostuu puurakennemuotoisesta sivustonavigaatiosta, yhteystiedoista ja linkeistä (kuva 2). Alatunnisteen navigaatio mukautuu sen perusteella, onko käyttäjä kirjautunut sisään vai ei. Samaa alatunnistetta käytetään kaikilla julkiosan sivuilla.



Kuva 2. Voimanainen-verkkosivun julkiosan etusivun rakenne.

Julkiosan sisältösiivuja ovat kurssi-info, lisätiedot, itse kurssin verkkokauppasivu ja sisäänkirjautuminen. Kurssi-infossa on perustietoa kurssista, ja sen tarkoitus on antaa hyvä kuva siitä, mitä mahdollinen asiakas on ostamassa. Lisätiedoissa on kaikki pääosin lajittelematon tieto, kuten mahdolliset yhteistyökumppanit ja mahdolliset lähteet Savimäen kurssimateriaaleihin. Verkkokauppa, tai pikemminkin rekisteröintisivu, koostuu käyttäjäehdoista, jotka tulee hyväksyä, ennen kuin voi edetä, käyttäjän omista tiedoista ja itse ostopainikkeesta, josta pääsee maksamaan tuotteen.

WordPress-teema koostuu suppeimmillaan indeksisivusta (*index.php*), tyylietiedostosta (*style.css*), silmukasta (*loop.php*) ja ylä- (*header.php*) ja alatunnisteesta (*footer.php*) sekä mahdollisesti myös sivupalkista (*sidebar.php*). Näitä käytetään yleensä siten, että kaikki tiedostot kerätään indeksisivulle seuraavasti: Indeksitiedosto kutsuu ylä- ja alatunnistetta, silmukkaa ja lopuksi alatunnistetta. Ylä- ja alatunnisteen mukana tulee tyylietiedosto, ja silmukan mukana tulevat kaikki sivustolle kirjoitetut post-sivut. Lisäksi WordPress-teemaan usein kuuluu funktiotiedosto (*functions.php*) ja mahdollisesti JavaScript-tiedostoja ja mallipohjatiedostoja, joilla voidaan muun muassa muokata sivukohtaista rakennetta. [10.]

Voimanainen-sivusto rakennettiin kokonaan uudelle WordPress-teemalle. Teema rakennettiin siten, että tavallisen WordPressin *post*-sivut näyttävän etusivun tilalle vaihdettiin staattinen etusivu (*front-page.php*), jolla etusivu esitetään. Tämä staattinen sivu saa ylä- ja alatunnisteen (*header.php*) ja alatunnisteen (*footer.php*), aivan kuten kaikki muutkin julkiosan sivut. Sivut näytetään joko sivu (*page.php*) -tiedoston rakennetta noudattaen tai vaihtoehtoisesti sivukohtaista mallipohjatiedostoa käyttäen, kuten esimerkiksi sisäänkirjautumiseen käytetyllä sivulla (*login.php*).

Etusivu on staattinen sivu. Sen sisältö tulee sille määrätystä WordPress-sivusta ”Etusivu”, mutta koska etusivulla on kaksi erillistä tekstikenttää, jouduttiin käyttämään erikoisratkaisua. WordPressillä on hankalaa, ellei jopa mahdotonta, erotella tekstiä useampaan osaan, mikäli tekstin sekaan ei kirjoiteta HTML-ohjelmointikieltä. HTML:n käyttäminen post-sivuilla tulisi kuitenkin välttää, sillä se todennäköisesti sekavoittaa ylläpitäjän tehtävää. Tästä syystä etusivu käyttää muokattavia kenttiä (*Custom Fields*). Etusivulla kutsutaan muokattavaa kenttää, jolle on annettu nimeksi ”Kurssikuvaus”. Kentän arvon voi asettaa WordPressin hallintapaneelisti, etusivun ominaisuuksista, ja siten etusivulle voitiin luoda myös toinen tekstikenttä juuri siihen, mihin se haluttiinkin [11]. Etusivu kokonaisuudessaan rakenteen suhteen ja muokattavien kenttien käyttö näkyvät kuvassa 3.



Kuva 3. Voimanainen-sivuston etusivun rakenne.

Julkiosan alatunniste pääosin luotiin WordPressin omaa dynaamista sivupalkkia käyttäen. Dynaamista sivupalkkia (*Dynamic Sidebar*) kutsutaan seuraavalla tavalla: "*dynamic\_sidebar( 'footer-sidebar' )*". Yksinkertaisesti funktio kutsuu sivupalkkia, joka on määritelty tunnisteella "footer-sidebar". Tämä tunniste on linkitetty hallintapaneelin Widgets-osiossa omaan säiliöönsä "Alatunnisteen tiedot", johon voidaan raahata mitä tahansa WordPressin pienoisohjelmia (widget), mikä on visualisoituna luvun 3.2 lopussa. Alatunniste sisältää näiden pienoisohjelmien lisäksi sivustonavigaation. [12.]



## Navigaatio

Julkiosaa voi navigoida kolmesta eri paikasta: ylätunnisteesta, alatunnisteesta tai alasivuilla olevasta sivukohtaisista dynaamisesta ylätunnisteesta. Ylätunnisteen navigaatio rakennettiin käyttämällä WordPressin omaa valikkofunktiota `wp_nav_menu` hyödyntäen, jota kutsuttaessa argumenteiksi annetaan tunnistenimi ja valikoissa näkyvä nimi sekä poistetaan ylimääräinen vakiona elementtiä ympäröivä säiliö. Tämä funktio luo WordPressin hallintapaneeliin mahdollisuuden luoda ja muokata valikolle sisältöä helposti. Samaa tekniikkaa käyttäen tehtiin myös alatunnisteen puurakennetyylinen sivukartta. Alatunnisteessa erona tosin on se, että siellä valikko on dynaaminen: valikon sisältö riippuu siitä, onko käyttäjä sisäänkirjautunut kurssilainen vai ei. [13.]

Sivukohtaista dynaamista ylätunnisteen navigaatiota ei ole luotu WordPressin omalla funktiolla, vaan täysin omalla funktiolla `vn_page_header_navigation`. Tämä funktio tekee tietokantakyselyn sivun tunnistetta argumenttina käyttäen ja tutkii, onko sivulla alasivuja. Tuloksista riippuen funktio tulostaa oikeat valikot, joista pääsee sivun yläsivuille.

## JavaScript

JavaScript on selaimen puolella suoritettava ohjelmointikieli, joka ei siten vaadi palvelinkutsua. Tämä tarkoittaa sitä, että JavaScript ei itsessään kuluta palvelimen resursseja eikä käyttäjän Internet-kaistaa. JavaScript on tehokas keino nopeuttaa verkkosivun toimintaa, parantaa käytettävyyttä ja usein myös tehdä sivustosta tyylikkäämpi [14]. Voimanainen-sivustolla JavaScriptiä käytetään niin julkiosassa kuin myös kurssin puolella.

Julkiosassa JavaScriptiä käytettiin esimerkiksi tarkistamaan käyttäjäehtojen hyväksyminen, missä muutokset käyttäjäehtojen tarkistuslaatikkoon otetaan kiinni jQueryn `change`-menetelmällä [15]. Mikäli muutos havaitaan, tarkistetaan, onko tarkistuslaatikko valittu vai ei, ja mikäli se on, näytetään käyttäjälle tilauslomake. Mikäli käyttäjä taas poistaa valinnan, piilotetaan lomake, jotta väärinkäytöksiä ei synny. Lisäksi JavaScriptillä onnistuttiin muuttamaan alatunnisteeseen tuotettavien eri `div`-elementtien jäsentely dynaamiseksi. Kun alatunnisteeseen tuotetaan lisää `div`-elementtejä, syntyisi ilman JavaScriptiä ongelmia rivittämisestä, koska kaikki rivit työntyisivät samaan säiliöön, mikä oli testattaessa ongelma. Sivusto ei tällä hetkellä käytä yli kolmea `div`-elementtiä

alatunnisteessa, mutta mikäli niitä halutaan jatkossa lisätä, ei siitä tule enää ongelmaa. JavaScriptin funktio *loopFooterWidgets* tunnistaa kaikki alatunnisteeseen luodut div-elementit tietyn tunnisteiden mukaan, tarkistaa niiden määrän ja luo niistä enimmillään kolmen div-elementin jonoja.

### Erilliset funktiot

WordPressissä sivuston omat funktiot tulee sijoittaa functions.php-tiedostoon, jotta WordPress osaa ne automaattisesti sieltä tunnistaa [10]. Voimanainen-sivustolla näin tehtiinkin, ja funktioita käytettiin niin julkiosan kuin myös kurssisivujen puolella. Julkiosassa funktioita tosin käytettiin niukasti, sillä WordPress itsessään tarjoaa niitä kattavasti. Julkiosan funktiosta esimerkkinä on *get\_voimanainen\_page*-funktio, joka ottaa argumenteikseen pakollisena tunnisteiden ja vapaaehtoisesti nimen. Funktio palauttaa argumenteista riippuen linkin. Argumenteista ensimmäinen, tunniste, määrää linkin osoittavan sivun osoitteen. Toinen argumentti, nimi, on oletuksena tunnisteiden osoittaman sivun otsikko, ellei sitä ole asetettu erikseen, jolloin annettua nimeä käytetään linkin tekstinä.

### Tyyli

Julkiosan tyyli määritellään CSS-tyylitiedostossa (*style.css*). Tässä tyylitiedostossa on hieman alle 900 riviä tekstiä, eikä sitä tässä raportissa käydä lävitse. Lyhyesti voi mainita, että sivusto käyttää laajalti hyödykseen CSS3:n ominaisuuksia, kuten esimerkiksi *Box Shadow* ja *Border Radius*. Lisäksi insinööriyössä käytettiin CSS:n vale-elementtejä (Pseudo-element) *:before* ja *:after*, ja lisäksi webkit-selainmoottorin ominaisuuksia, kuten *-webkit-font-smoothing* ja *-webkit-hyphens*. [16.]

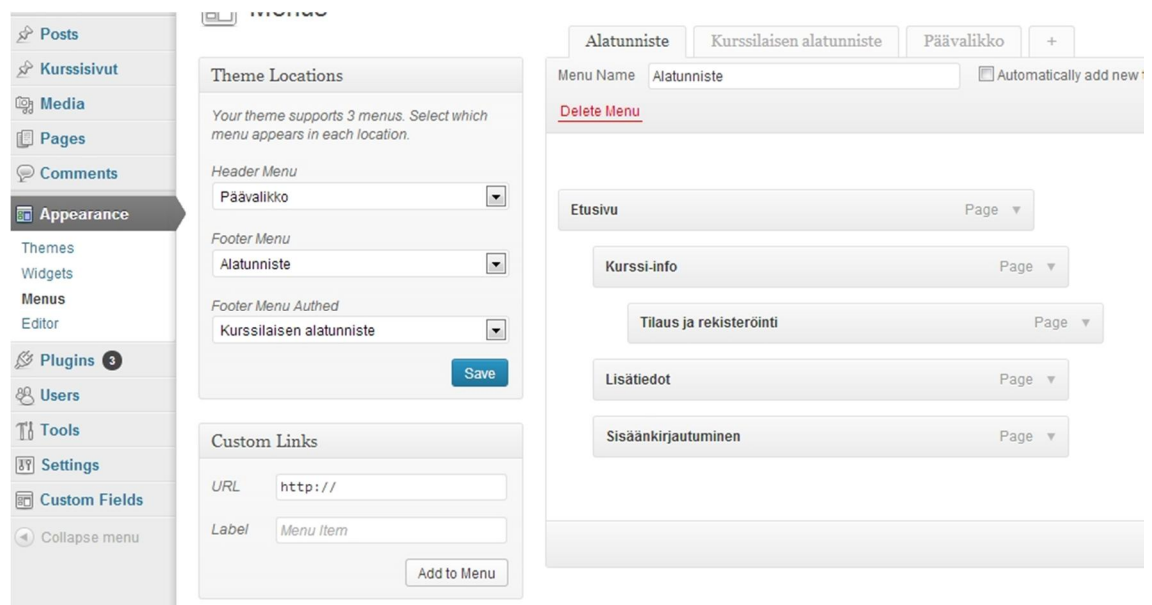
Näitä samoja CSS:n ominaisuuksia käytetään hyväksi myös kurssiosan CSS-tyylitiedostossa (*course.css*), eikä niitä siksi erikseen mainita kurssiosan esittelyssä.

### 3.2 Julkiosa käyttäjän ja ylläpitäjän näkökulmasta

Voimanainen-sivulle tultaessa käyttäjän pitäisi hyvin helposti saada selville, mistä sivustolla on kyse. Etusivu suunniteltiin siten, että turhia silmille hyppääviä elementtejä olisi mahdollisimman vähän ja että tärkeitä tietoja on korostettu kontrastin keinoin sekä

keskeisten elementtien asettelulla. Käyttäjän pitäisi myös suoraan pystyä siirtymään loogisesti kurssikuvausta tarkentavalle sivulle (Kurssi-info), eikä kurssin ostamisessa pyydetä enempää tietoa kuin käyttäjältä vähimmillään tarvitsee vaatia. Kurssisivulle siirtyminen onnistuu pääsivulta paristakin paikasta, joten sen pitäisi olla myös melko helppoa.

Ylläpitäjän, eli tässä tapauksessa insinööriyön asiakkaan, pitäisi pystyä muokkaamaan lähes kaikkia sivuston tekstikenttiä loogisesti. Tämä toteutettiin käyttäen pelkästään WordPressin omia ominaisuuksia. Sivut tehtiin loogisiksi käyttäen mallipohjasivuja aina silloin, kun tarve oli, minkä takia sivuston rakenne on juuri sellainen kuin oli suunniteltukin. Esimerkkinä sivuston muokattavuudesta voidaan käyttää alatunnistetta, jota voidaan muokata sisällön puolesta juuri sellaiseksi kuin ylläpitäjä haluaa. Tämä esimerkki on havainnollistettuna kuvassa 4, joka on kuvankaappaus hallintapaneelin kohdasta, jossa mainitun alatunnisteen puurakenteista sivustonavigaatiota muokataan.



Kuva 4. Voimanainen-julkiosan alatunnisteen navigaatio hallintapaneelissa.

Kuvassa 5 on vielä näyte siitä, miten alatunnisteen muuta sisältöä voidaan muokata.

ivate them and delete their settings.

Categories

A list or dropdown of categories

Meta

Log in/out, admin, feed and WordPress links

Recent Posts

The most recent posts on your site

Tag Cloud

Your most used tags in cloud form

Alatunnisteen tiedot

Tähän kaikki alatunnisteessa näytettävät tiedot ensimmäistä sivustokarttaa lukuunottamatta.

HTML Javascript Adder: Yhteystiedot

Title :

Yhteystiedot

Content :

<> Toolbar ? Help 🔍 Preview

Marja-Leena Savimäki

Kouluttaja, Muutosvalmentaja

NIP Trainer Social Panorama Trainer

Fellow Member Trainer IANIP

+358 50574 0080

marja[viiva]leena[at]savimaki.net

Settings

☒ Show in all pages

☐ Show only in specific pages

☐ Hide only in specific pages

☒ Automatically add paragraphs ☐ Don't display to admin

Delete | Close | Like | Donate !

Save

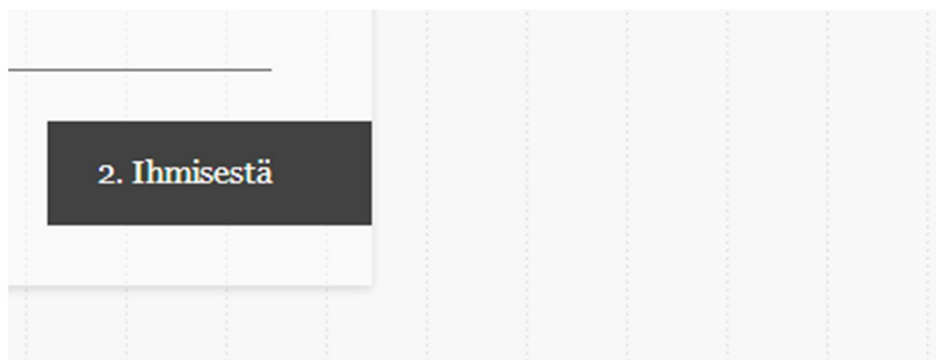
HTML Javascript Adder: Linkit

Kuva 5. Alatunnisteen sisältö hallintapaneelissa.

Alatunnisteeseen voidaan siis sijoittaa tekstiä ja jäsenellä sitä halutulla tavalla tai vaihtoehtoisesti listata linkkejä. Kuten luvusas 3.1 mainitaan, alatunniste käyttää WordPressin pienisohjelmia, minkä ansiosta alatunnisteeseen voidaan sijoittaa itse asiassa todella erimuotoista tietoa, kuten vaikka hakutoiminto tai käyttäjätilastoja.

#### 4 Voimanainen-kurssiosan toteutus

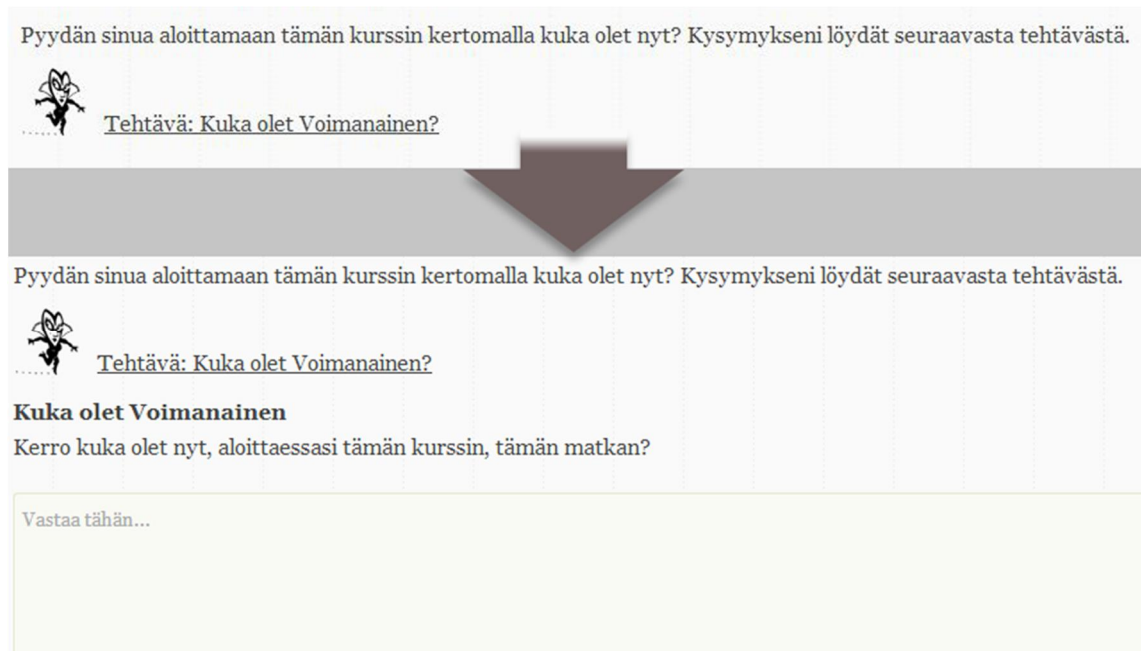
Kurssisivujen graafinen toteutus lähti suunnittelun puolella siitä ajatuksesta, että kurssisivut pidettäisiin mahdollisimman hillittyinä ja kaikkia raskaita elementtejä vältettäisiin muussa kuin korostuskäytössä parantamassa käytettävyyttä. Ihmisen silmä kohdistuu helposti raskaisiin osiin, ja tämän takia luettavuus heikkenee, mikäli raskaat osat ovat vain tyylikeino. Sivustolla onkin paljon helpompi navigoida, jos silmä ei jatkuvasti hypi epäolennaisesta elementistä toiseen. Tämä ajatus toteutettiin vaihtamalla koko kurssiosan teema julkiosasta eriäväksi. Taustalle tuli vaalea hillitty liituroitakuvio ja sen päälle sisältöä rajaamaan puolittain läpinäkyvä valkoinen elementti (kuva 6). Tämä pohjaratkaisu tarjosi hyvät mahdollisuudet hyödyntää kontrastieroja pohjan suhteen vaaleanpunaiseen ja lähes mustaan väriin (kuva 6). Kurssisivujen väriteemaksi muodostui siten valkoisen, vaaleanpunaisen ja tumman harmoninen yhdistelmä, jota käytettiin esimerkiksi kaikissa kurssisivujen painikkeissa: lähtökohtaisesti painikkeen teksti on valkoinen ja tausta tumma, joka hover-efektissä vaihtuu vaaleanpunaiseksi tekstin pysyessä valkoisena. [17.]



Kuva 6. Kurssisivuston väriteemassa käytetty tumma väri ja liituroita.

Etusivulla on ylätunniste, joka kattaa navigaation, ja itse sisältöosa, jonka sisältö muuttuu dynaamisesti sen mukaan, onko käyttäjä aloittanut jo kurssin vai ei. Mikäli käyttäjä ei ole vielä aloittanut kurssia, hänen tulee hyväksyä käyttäjäehdot, jotka etusivulla näytetään. Kun ehdot on hyväksytty, käyttäjä pääsee aloittamaan kurssin painiketta painamalla. Käyttäjätehtöjen lisäksi käyttäjälle näytetään ohjeita kurssin suorittamiseksi ja kerrotaan sen ominaisuuksista. Ohjeita ja ominaisuuksia pääsee käyttämään kurssin aloituksen jälkeenkin. Etusivulta poistuvat käyttäjäehdot sen jälkeen, kun kurssi on aloitettu, ja lisäksi käyttäjää ei enää kehoiteta aloittamaan kurssia, vaan jatkamaan sitä.

Itse kurssin sisältösivut ovat yksinkertaiset ja antavat sisällölle tilaa. Sisältö jäsenneltiin insinööriyön asiakkaan haluamalla tavalla, ja kurssitehtävät automaattisesti piilotettiin painikkeen taakse, jotta ne eivät veisi tilaa, ennen kuin niiden täyttäminen on ajankoh- taista. Kuvassa 7 ovat esitettynä kurssitehtävät ensin piilotettuna ja sitten avattuna. Kurssisivulle kehitettiin myös kirjanmerkki, jolla käyttäjä voi helposti merkitä edistyk- sensä sivulla, tallentaa sen samalla, kun tallentaa koko kurssin edistyksensä, ja jatkaa seuraavalla kerralla siitä, mihin jäi.

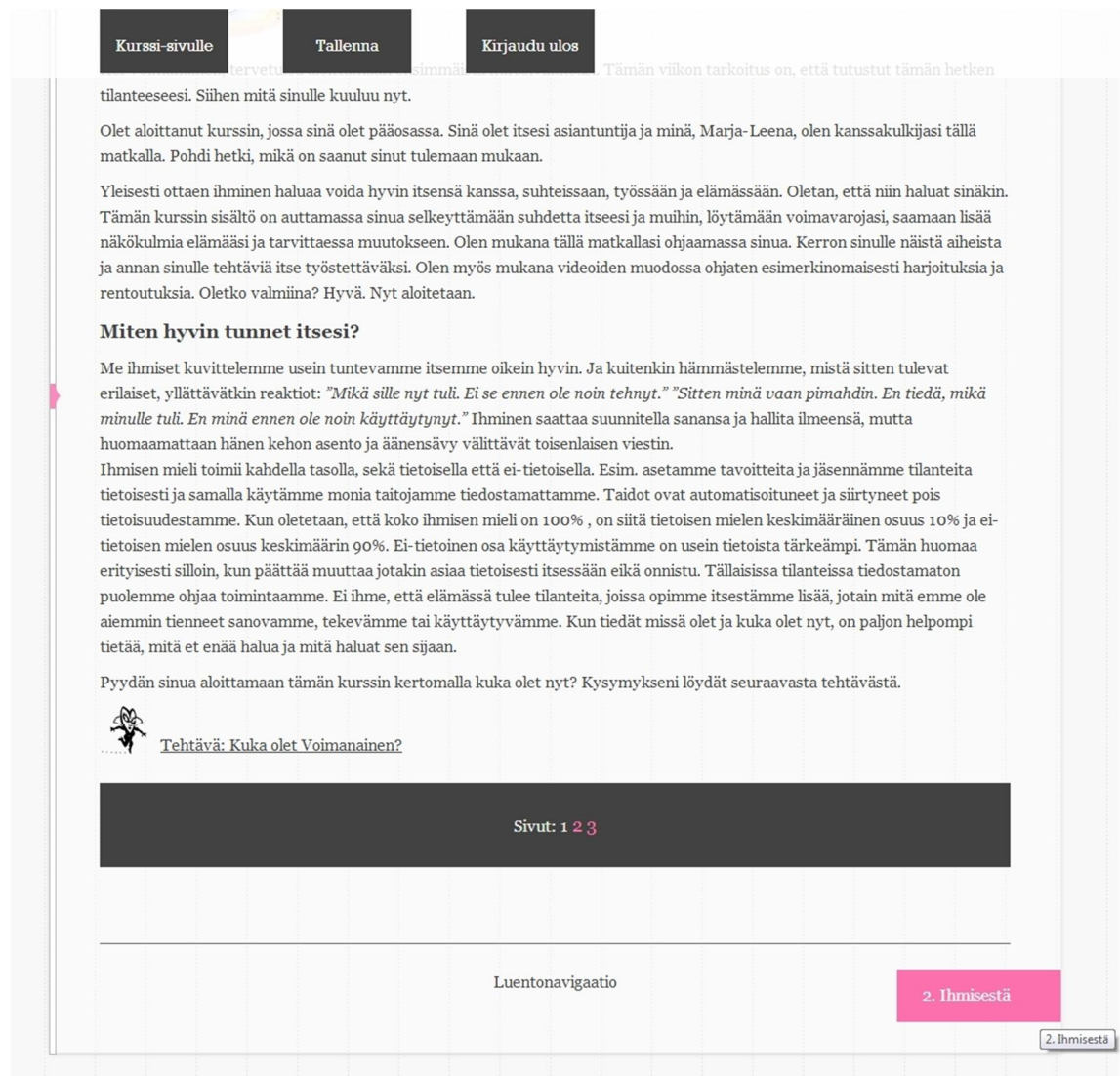


Kuva 7. Tehtävien näyttäminen painikkeen takana.

Kurssiviikkojen tehtävien tallentaminen oli suunnittelussa yksi tietoturvan pääkysymyk- siä. Tehtävien tallentamisen tärkeyttä pohdittiin pitkään, ja siitä päätettiin ainakin tässä vaiheessa luopua. Käyttäjä ei siis pääse mitenkään tallentamaan sivustolle vastauksi- aan, mutta hänellä kuitenkin on mahdollisuus tallentaa sivusto omalle tietokoneelleen tai vaihtoehtoisesti tulostaa vastauksensa paperille, jota helpottamaan sivustolle tehtiin oma painikkeensa JavaScriptillä. Alun perin kurssiviikkojen tulostus oli tarkoitus onnis- tua PDF-muotoon, mutta tätä tapaa ei saatu toimimaan muilla kuin Googlen Chrome- selaimella, joten siitä päätettiin luopua. Ongelmalliseksi PDF:n luomisen teki se, että käyttäjän vastaukset olisi pitänyt tallentaa tietokantaan, jotta PDF:n luominen vastauk- sien kanssa olisi onnistunut PHP:lla. Ainoa vaihtoehto luoda kurssinäköymästä PDF, jossa näkyisivät myös vastaukset, olisi ollut käyttää JavaScriptiä, mutta kuten aiemmin

mainittiin, se aiheutti suuria selainongelmia. Tästä lisää teknistä toteutusta käsittelevässä luvussa 4.1 (s. 20).

Kuvassa 8 on ensimmäisen viikon luentosivu kuvankaappauksena. Kuvassa näkyvät ylhäällä navigaatiopainikkeet, vasemmassa laidassa kirjanmerkki, keskellä tavallinen tekstisisältö, kurssin tehtäviin johtava linkki tekstiosuuden alla ja sen alla kurssiviikon muut sivut. Alimmaiseksi on eroteltu luentonavigaatio, josta pääsee seuraavalle viikolle. Kuvan oikeassa alakulmassa näkyy myös painikkeen leijuntaefekti, jota käytetään kaikissa sivuston painikkeissa.



Kuva 8. Erään kurssiviikon kurssisivu.

Kuvassa 8 näkyvä tehtäviin johtavan linkin edessä oleva hahmo ei ole tässä insinööri-työssä suunniteltu, vaan ulkoa tullutta materiaalia, kuten luvaussa 2.5 (s. 7) mainittiin.

#### 4.1 Kurssiosan tekninen toteutus

Kurssiosa rakentuu kurssin etusivusta (*course.php*), sen ylä- (*course-header.php*) ja alatunnisteesta (*course-footer.php*) sekä kurssiviikkojen sivuille rakennetusta mallipohjasivusta (*single-vn\_coursepage.php*). Kurssisivu toimii etusivuna, ja sen sisältö riippuu siitä, onko käyttäjä aloittanut kurssin suorituksen vai ei. Lisäksi kurssisivuille ei pääse, mikäli ei ole kurssia ostanut tai ylipäänsä rekisteröitynyt.

Voimanainen-sivusto käyttää erikseen lisättyä käyttäjäroolia nimeltä "Kurssilainen". Kurssilaisella on lukuoikeudet ja view-course-oikeus, jota tarkastellaan jokaista kurssisivua ladattaessa. Tähän käytetään hyväksi WordPressin omaa funktiota, jolla tarkastellaan sisäänkirjautuneen käyttäjän käyttöoikeuksia seuraavalla tavalla: "*current\_user\_can( 'view-course' )*". Mikäli käyttäjällä on oikeudet kurssiin, funktio palauttaa tosi-arvon, ja mikäli käyttäjällä taas ei ole oikeuksia kurssiin, häntä pyydetään joko kirjautumaan sisään tai ostamaan kurssi. [18.]

Kurssin etusivun rakenne koostuu ylä- ja alatunnisteesta ja niiden väliin jäävän sivun (Kurssisivu) sisällöstä, joka haetaan ja tulostetaan, mikäli käyttäjä ei ole vielä aloittanut kurssia. Käyttäjää pyydetään hyväksymään kurssin ehdot rastittamalla lomakkeen valintaruutu, minkä jälkeen kurssin pääsee aloittamaan aloituspainikkeesta. Aloitus-painike on lomakkeen sisällä oleva submit-painike. *Submit* otetaan kiinni POST-menetelmällä ja samassa tarkastetaan, että myös valintaruutu on rastitettu. Mikäli kumpikin ehto täyttyy, kutsutaan funktiota *vn\_start\_course*, joka merkitsee kurssin aloitetuksi, minkä jälkeen kutsutaan WordPressin omaa funktiota *wp\_redirect*, joka saa argumenttikseen ensimmäisen kurssiviikon linkin, jonne käyttäjä sitten siirretään.

Mikäli sisäänkirjautunut käyttäjä on jo aloittanut kurssin, hänelle näytetään kurssin jäljellä oleva suoritus aika, suoritusohjeet ja painike, josta käyttäjä pääsee jatkamaan kurssia. Kurssin jatkamiskohta määritellään tietokannasta saatavalla tiedolla *course\_progress*-avaimen alta, joka yhdistetään kurssisivun verkko-osoitteeseen, josta muodostuu kokonainen linkki. Tietokantaan tallennetusta *course\_progress*-avaimesta lisää Funktiot- ja JavaScript-osioissa.



Kurssin ylätunnisteessa on yksinkertainen navigaatio, joka toteutettiin tavallisella järjestämättömällä listalla. Listan poikkeukselliset kohdat ovat asetukset- ja tallenna-toiminto. Ne vaihtelevat keskenään sen mukaan, ollaanko kurssin etusivulla vai jonkin kurssiviikon sivulla. Asetukset-toiminto näkyy vain kurssin etusivulla, ja se paljastaa yksinkertaiset lomakkeeseen upotetut asetukset, joista käyttäjä pääsee vaihtamaan salasanan ja sähköpostiosoitteen. Kun käyttäjä syöttää esimerkiksi uuden salasanan, sitä tarkastellaan ensin JavaScriptillä, ja jos salasana on hyväksyttävää muotoa, se lähetetään POST-menetelmällä PHP:lle. Uusi salasana tallennetaan tietokantaan käyttäen WordPressin omaa `wp_set_password`-funktia. WordPressillä on myös oma hallintapaneeli näille toiminnoille, mutta sitä ei haluttu käyttää, sillä se olisi rikkonut pahasti sivuston teemaa.

Kurssiviikon sivulla näytettävä tallenna-ominaisuus on itse asiassa kokonainen lomake, jonka arvot ovat piilotettuja. Lomakkeen piilotetut syötteet, *top-val* ja *slug-val*, saavat arvonsa tietokannasta tai JavaScriptistä. Ensimmäisenä mainittu arvo, *top-val*, on oletuksena tietokannasta tuleva kirjanmerkin paikkaa esittävä liukuluku, joka korvataan välittömästi JavaScriptistä tulevalla arvolla, mikäli kirjanmerkkiä aletaan liikuttaa. Jälkimmäinen syöte, *slug-val*, saa arvokseen juuri silloiseen kurssiviikon sivuun osoittavan verkko-osoitteen loppuosan, joka asetetaan automaattisesti JavaScriptin toimesta aina, kun sivu ladataan. Verkko-osoitteen loppuosalla tarkoitetaan tässä tapauksessa kahden tai kolmen viimeisen vinoviivan sisään jäävää osaa. Esimerkiksi ensimmäisen kurssiviikon ensimmäisen sivun verkko-osoite on muotoa `"http://www.voimanainen.net/kurssisivu/1-unelmat-toiveet-haaveet/"`, jossa loppuosa on vain `"1-unelmat-toiveet-haaveet/"`, joka tallennetaan *slug-val*-syötteen arvoksi. Tämän lisäksi jokaisen kurssiviikon seuraavat sivut näkyvät verkko-osoitteen perässä, eli edellämmainitun ensimmäisen kurssiviikon toisella sivulla muodostuisi *slug-val*-syötteen arvoksi `"1-unelmat-toiveet-haaveet/2/"`. Tallenna-painiketta painettaessa lomakkeen tiedot otetaan PHP:ssa vastaan POST-menetelmällä ja tallennetaan tietokantaan käyttäjän metatiedoksi.

Kurssiviikon sivut on rakennettu WordPressin *Custom Post Type* -sivutyyppiä käyttäen. Jokainen kurssiviikko on oma *custom post* -sivunsa, joka tunnistetaan nimellä *vn\_coursepage*, mikä mahdollistaa sen, että kurssiviikon sivuja voidaan automaattisesti kysellä niiden mallipohjasivulla (`single-vn_coursepage.php`). Tiedostonimen alussa oleva "single" viittaa siihen, että kysellään vain yhtä sivua kerralla, ja sitä seuraava

osa viittaa loogisesti siihen, että halutaan kysellä vain tämän tunnisteeseen sisään kuuluvia sivuja. [19.]

## Funktiot

Ensimmäisenä kurssin pääsivulle tultaessa tarkastellaan, onko käyttäjä aloittanut jo kurssin. Tämä tapahtuu kutsumalla funktiota *vn\_course\_started*, joka ottaa argumentikseen käyttäjän tunnistenumeron. Funktio yksinkertaisesti tekee WordPressin oman funktion *get\_user\_meta* avulla kyselyn tietokannassa olevista metatiedoista avaimella "timeleft" käyttäen saamaansa tunnistenumeroa. Mikäli tietokannasta saadaan muuta kuin tyhjä kenttä, voidaan todeta, että kurssi on aloitettu, missä tapauksessa funktio palauttaa tosi-arvon. Mikäli funktio kuitenkin palauttaa epätoden, on kyseessä uusi käyttäjä, ja jos käyttäjä päättää tämän jälkeen aloittaa kurssin, kutsutaan funktiota *vn\_start\_course*, joka tallentaa tietokannan metatietoihin WordPressin funktiota *update\_user\_meta* käyttäen uuden avaimen "timeleft", jonka tunnistenumeroiksi tulee käyttäjän tunnistenumero. Uusi avain saa arvokseen päivämäärän, joka on tasan neljän kuukauden päässä senhetkisestä päivästä.

Kun kurssi on aloitettu, näkyy jäljellä oleva suoritus aika kurssin etusivulla. Tämä toteutettiin kutsumalla funktiota *vn\_timeleft*, joka tarkastelee tietokantaan tallennetun "timeleft"-avaimen arvoa ja vertaa sitä senhetkiseen päivämäärään. Funktio palauttaa erotuksen, mikäli se on yli nolla. Jos erotus on kuitenkin pienempi tai yhtä suuri kuin nolla, se tarkoittaa, että kurssin suoritus aika on päättynyt. Tässä tapauksessa käyttäjätunnus poistetaan tietokannasta, kuten kurssin käyttöehdoissa on ilmoitettu.

Kurssin suorituksen edistymisen tallennukseen käytetään funktiota *vn\_update\_progress*, joka saa argumenteikseen käyttäjän tunnistenumeron, verkko-osoitteen ja kirjanmerkin etäisyyden sivun yläreunasta pikseleinä mitattuna. Funktio tallentaa nämä arvot käyttäjän metatietona tietokantaan omien avaimiensa alle. Myös tämä tallennusprosessi käyttää hyväkseen WordPressin omaa *update\_user\_meta*-funktiota. Kun kurssia jatketaan, haetaan tietokantaan tallennettu "course\_progress"-metatiedon arvo funktiolla *vn\_course\_progress*. Funktion palauttava muuttuja on vain merkkijono, joka sisältää, kuten jo aiemmin mainittiin, verkko-osoitteen loppuosan, joka kurssin etusivulla liitetään sen omaan verkko-osoitteeseen. Sivuston kirjanmerkin paikka taas haetaan *vn\_get\_bookmark*-funktiolla, joka vastaavasti hakee tietokannasta kirjanmerkin sijainnin. PHP toimii tässä tapauksessa vain tallentimena ja tietokannasta

tiedon hakijana. Itse tiedon päivitys käyttäjän syötteistä riippuen tapahtuu JavaScriptillä.

## JavaScript

Kurssisivuston puolella JavaScriptiä käytetään huomattavasti enemmän kuin julkiosan puolella. Ensimmäisenä esittelen kirjanmerkin toteutusta, johon luvussa ”Funktiot” edellä PHP:n osalta jäätin. Kirjanmerkkiä liikutellaan hiirellä, jota JavaScript-kirjasto jQueryn mousedown- ja mousemove-menetelmillä seurataan. Kummallakin seurataan pystyakselin liikettä, päivitetään sitä itse kirjanmerkin div-elementtiin ja samaan aikaan myös, jo aiemmin mainitun, tallennuslomakkeen top-val-syötteen arvoksi. Käyttäjän liikuttaman kirjanmerkin paikka päivittyy siis jatkuvasti tallennettavan syötteen arvoksi, joten kurssin suoritus kirjanmerkin puolesta tallentuu aina tarkasti ajasta riippumatta.

Kurssin suoritus halutaan tallentaa myös sen perusteella, mille sivulle käyttäjä on edellisellä kerralla jäänyt. Tämä hoituisi PHP:lla, jos jokaisella sivulla olisi oma tunnistenumeronsa, mutta näin ei ole, sillä sivusto käyttää WordPressin omaa tapaa (<!--nextpage-->) jakaa yksittäinen kurssiviikon sivu osiin ja tätä tapaa käyttäessä eri sivuille ei synny omia tunnisteita, vaan ne erotetaan vain DOM:ssa, johon PHP:lla ei ole rajapintaa. Tästä syystä tieto eri sivuista saadaan verkko-osoitteesta, jonka arvoa JavaScriptillä voidaan tutkia ja joka voidaan tallentaa haluttuun muuttujaan komennolla *window.location*. Tässä tapauksessa tämä ei kuitenkaan tuota haluttua tulosta, sillä verkko-osoitteesta halutaan tallentaa vain se osa, joka tulee kurssin etusivun jälkeen. Ongelma korjataan siten, että kutsutaan *window.location.pathname*, katkaistaan siitä saatava merkkijono kohdasta ”kurssisivu/” ja tallennetaan tästä jäljelle jäävä osa omaan muuttujaan. Tämä muuttuja tallennetaan ”slug-val”-syötteeseen joka kerta, kun sivu ladataan. Tällä tavoin kurssilla edistyminen tallentuu aina oikein. [20.]

JavaScriptiä käytetään kurssin sivuilla eri elementtien piilottamiseen ja taas esittämiseen, kuten kurssitehtävien ja etusivun ohjeiden kanssa. Tämä tehdään yksinkertaisella CSS-luokan poistamisella ja lisäämisellä, jotka onnistuvat jQueryn *addClass*- ja *removeClass*-metodeilla. Näiden lisäksi JavaScriptillä onnistuttiin hallitsemaan kurssiviikkojen sivuilla esiintyviä seuraavalle ja edelliselle kurssiviikolle johtavien linkkien sisältöjä. Ongelmana olivat aluksi liian pitkät linkit, jotka johtuivat kurssiviikoille annetuista pitkistä nimistä. Nämä liian pitkät linkit rikkoivat sivuston rakennetta ja tyyliä pahasti. Ongelma ratkaistiin käyttämällä JavaScriptiä linkkien tekstiosien pituuksien tarkkailuun:

mikäli teksti on yli 25 merkkiä pitkä, korvataan sen yli menevä osa kolmella pisteellä. Tämä ratkaisu on havainnollistettu esimerkkikoodissa 1.

```
if ( nextLink.text().length > 25 ) {

    var text = nextLink.text();
    var shortlink = text.substring(0, 25)+"...";
    nextLink.empty();
    nextLink.attr('title', text)
    nextLink.append(shortlink);
}
else{
    var text = nextLink.text();
    nextLink.attr('title', text)
}
```

Esimerkkikoodi 1. JavaScriptin käyttö linkkien lyhentämiseen.

Aiemmin luvussa 4 mainittiin, että kurssiviikon sivujen tulostus PDF:ksi haluttiin mahdollistaa ja että tämän ominaisuus pyrittiin toteuttamaan JavaScriptillä, mikä kuitenkin ei toiminut. Tätä ominaisuutta varten tehtiin JavaScriptillä jQueryä hyödyntäen keyup-tapahtumafunktio. Funktio tutkii jokaisen tehtävien vastaukseen käytettävän textarea-elementin sisällä tapahtuvan näppäimistön painalluksen, ottaa siitä syntyvän kirjainyhdistelmän talteen ja liittää sen textarea-elementin sisällöksi. Käytännössä siis funktio kirjoittaa DOM:iin kaiken, mitä käyttäjäkin kirjoittaa. Ilman tätä funktiota PDF:n tuottaminen JavaScriptillä ei olisi onnistunut. Funktio päätettiin jättää toiminnalliseksi, sillä se ei tee sivustosta juuri sen raskaampaa ja sitä voidaan hyödyntää tulevaisuudessa, mikäli PDF:n tulostus JavaScriptillä otetaan joskus uudelleen käyttöön.

JavaScriptillä tehtiin myös hyvin pieni funktio, joka antaa jokaiselle tehtävän textarea-elementille paikkamerkin (Placeholder), jonka arvoksi tulee kirjainyhdistelmä "Vastaa tähän...". Tällä tavoin ylläpitäjän ei tarvitse itse antaa sitä joka kerta, kun luo sivustolle tehtävän vastausalueen.

## 4.2 Kurssisivusto käyttäjän näkökulmasta

Kurssisivun toteutuksessa käyttäjän näkökulmaa ajateltiin jatkuvasti. Käytettävyys pyrittiin tekemään mahdollisimman hyväksi, ilman että ominaisuuksista luovuttaisiin, mikä aiheuttikin teknisiä haasteita. Esimerkkinä tästä kirjanmerkin toteutus, joka on käyttäjää helpottamaan suunnattu ominaisuus mutta jonka toteuttaminen teknisesti tuotti kuitenkin odotettua enemmän haastetta.

Käyttäjän toimintoja pyritään ohjaamaan muun muassa kontrastin keinoin mahdollisimman tehokkaasti, jotta hänen ei tarvitsisi harhailla sivustolla löytääkseen mitä tarvitsee, minkä toivotaan johtavan miellyttävään käyttäjäkokemuksesta. Käyttäjälle tarjotaan ohjeita kurssin suorittamiseen ja kerrotaan kurssin ominaisuuksista etukäteen. Ohjeisiin ja kurssin ominaisuuksista kertovaan informaatiopakettiin on pääsy myös helposti jälkikäteen. Lisäksi kurssin suorittamista on pyritty selkeyttämään korostamalla sen navigaatiota ja muita tärkeitä toimintoja. Esimerkkinä voidaan mainita pienet kuvakkeet ja alleviivauksen käyttö korostuksena jokaisen tehtäväosion linkin kanssa.

Kurssia suorittamaan siirtyneellä käyttäjällä on kurssin suorittamisen lisäksi pääsy myös muutamaan ominaisuuteen, jotka on tehty auttamaan käyttäjää muistamaan, mihin kohtaan kurssin suorituksessa on jäänyt edellisellä kerralla, mikä helpottaa kurssin suorittamista. Näitä ominaisuuksia ovat kurssiedistyksen tallentaminen ja kirjanmerkki.

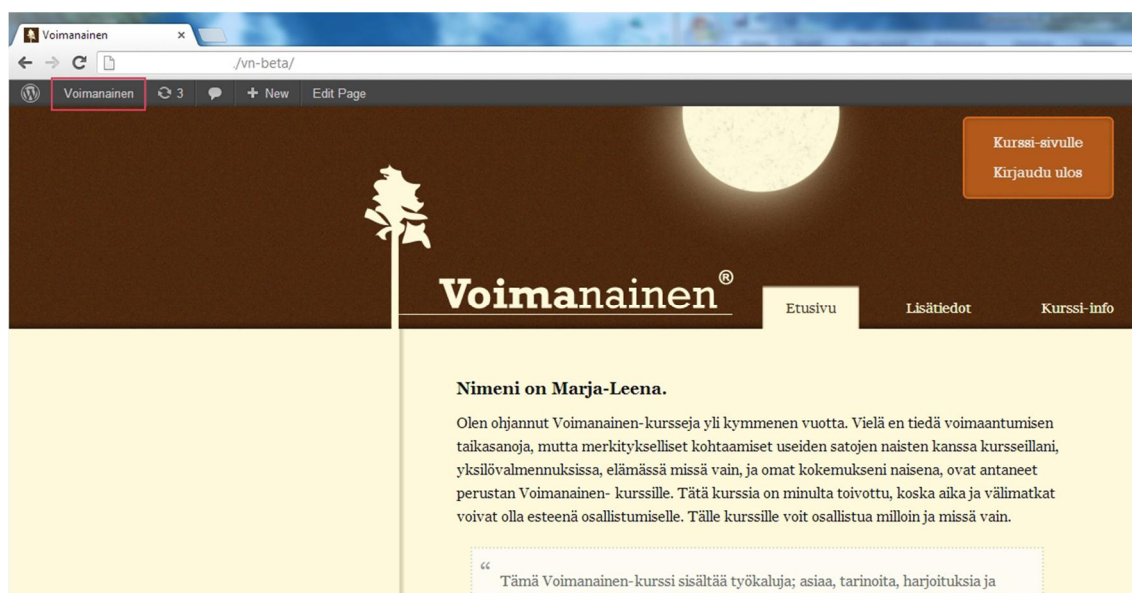
Kurssiedistyksen tallentaminen tallentaa muistiin sen sivun, jolla tallennus-painiketta painetaan. Kun käyttäjä saapuu jatkamaan kurssin suorittamista, hän pääsee kätevästi jatkamaan suoraan siitä, mihin on edistyksen tallentanut. Kirjanmerkki toimii saman tallennus-painikkeen avulla, eli kirjanmerkki siis tallentuu samalla kun kurssiedistyskin. Mikäli kirjanmerkkiä ei ole erikseen asetettu ennen tallennusta, se jää perusasentoonsa eli sivuston yläreunaan.

## 4.3 Ylläpitäjän näkökulma

Ylläpitäjän, eli tässä tapauksessa insinööritoimiston asiakkaan, näkökulmaa on toteutuksessa ajateltu siten, että hänen pitäisi pystyä jäsentelemään kurssin sisältöä juuri niin kuin haluaakin, pystyä muokkaamaan sitä jälkikäteen, lisäämään osioita ja vielä tekemään

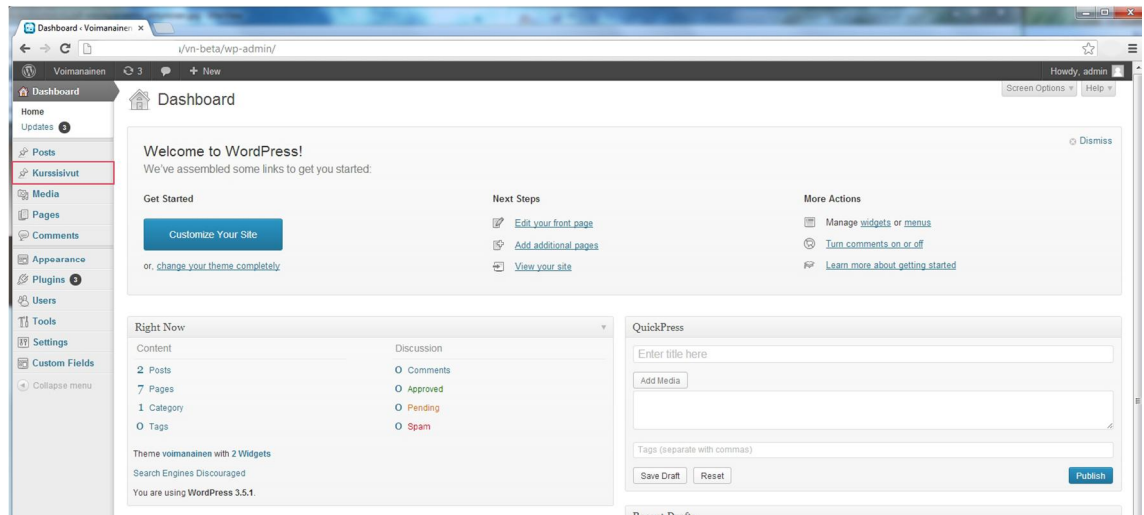
tämän kaiken helposti. Tämän toteutukseen käytettiin WordPressin ominaisuuksia mahdollisimman tehokkaasti. Esimerkiksi kurssiviikot toteutettiin, kuten jo teknistä toteutusta käsittelevässä luvussa kerrottiin, WordPressin omaa *Custom Post* -ominaisuutta käyttäen, mikä merkitsee sitä, että kaikki kurssiviikon sivut ovat muusta sivuston sisällöstä erillään. Tämä parantaa hallintaominaisuuksia ja selkeyttää hallintapaneelin navigointia. [19.]

Tämän luvun alussa mainittu helppo ylläpito havainnollistetaan seuraavilla kuvilla (kuvat 9-12). Kuvat antavat selkeän kuvan siitä, miten helposti ylläpitäjä voi kurssiviikkoja muokata sisäänkirjautumisen jälkeen. Kuvasarjan ensimmäisessä kuvassa 9 on julkiosan näkymä, jossa on punaisella ääriiviivalla korostettuna vain ylläpitäjälle näkyvä hallintapaneelin painike.



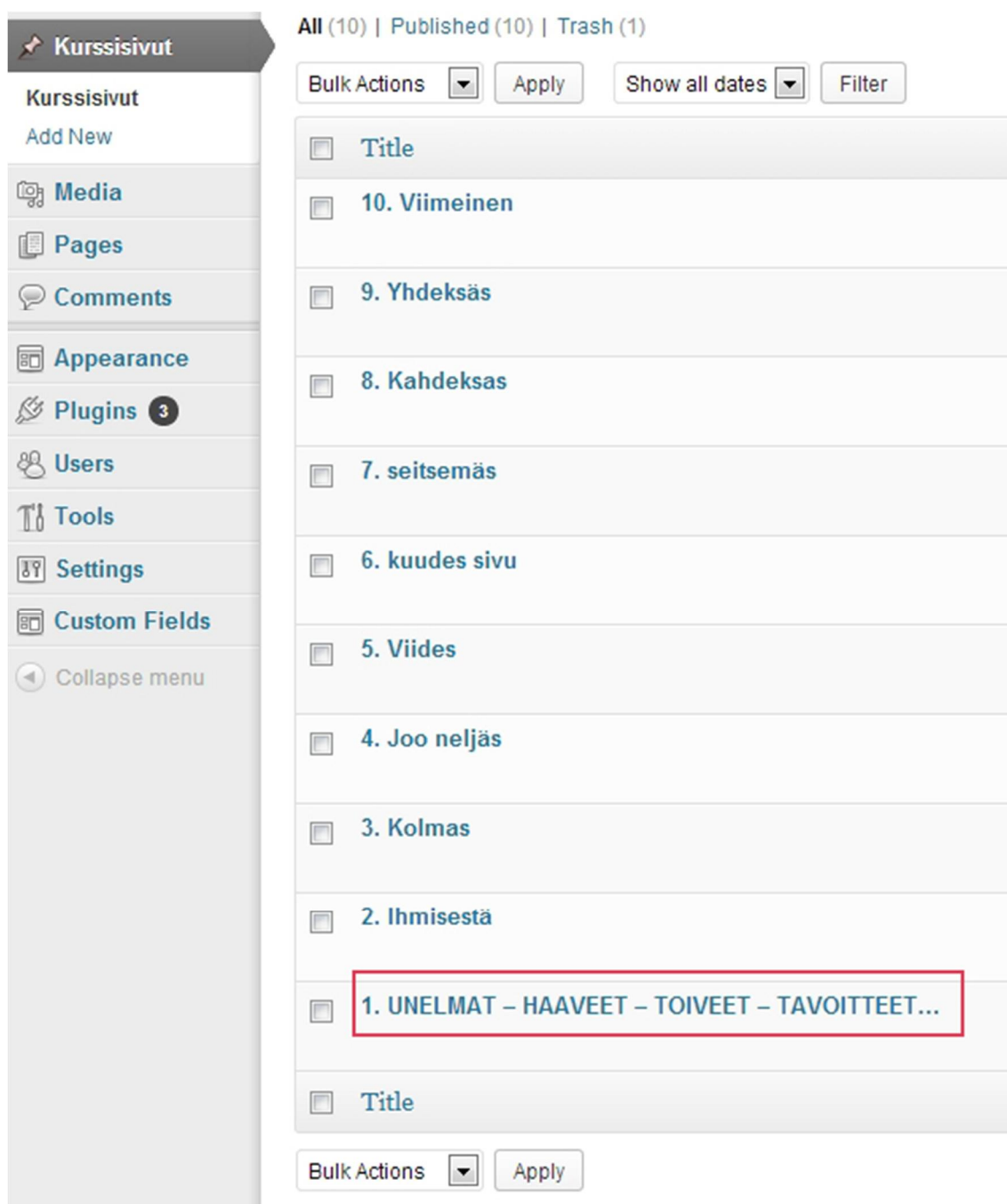
Kuva 9. Hallintapaneeliin siirtyminen korostettu punaisella suorakulmiolla.

Kuvassa 10 ylläpitäjä on siirtynyt hallintapaneeliin. WordPressin hallintapaneelissa navigaatio on vasemmassa laidassa, jossa tässä tapauksessa on punaisella laatikolla korostettuna "Kurssisivut"-osio.



Kuva 10. Hallintapaneelin päänäköymä.

Kuvassa 11 ylläpitäjä on siirtynyt "Kurssisivut"-osioon. Siinä ovat listattuna kaikki verkkokurssin kurssiviikkosivut. Korostettuna on kurssin avausviikko.



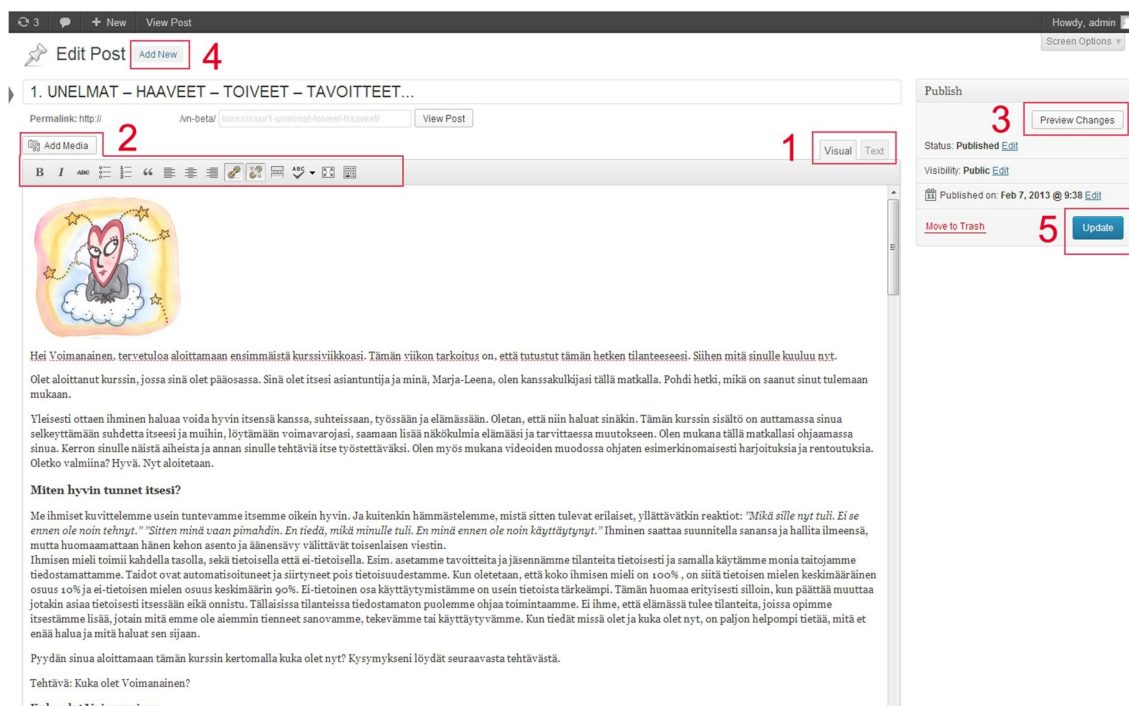
Kuva 11. Kurssisivut-näkymä.

Kuvassa 12 ylläpitäjä on siirtynyt muokkaamaan ensimmäistä kurssiviikkoa. Ylläpitäjälle aukeava näkymä on niin kutsuttu visuaalinen näkymä, joka kuvastaa sitä, miltä kurssi suurin piirtein näyttää muotoiluineen. Numerolla 1 on korostettuna valinta, katsellaanko visuaalista näkymää vai ohjelmointipuolen näkymää, johon voidaan myös kirjoittaa HTML-kielellä. Numerolla 2 korostetaan WordPressin käyttämää avoimen lähdekoodin JavaScriptia käyttävää HTML WYSIWYG -tekstimuokkainta TinyMCE. TinyMCE



on maailman eniten käytetty WYSIWYG-tekstimuokkain, jota käyttävät suurimmatkin yritykset. [21; 22.]

Numerolla 3 on korostettu *Preview Changes* -painiketta, jolla ylläpitäjä voi esikatsella tekemiään muutoksia ennen niiden julkaisua. Näin vältetään mahdollisten virheiden näyttämistä asiakkaille. Lisäksi numerolla 4 on korostettu uuden kurssiviikkosivun luontia ja numerolla 5 muokattavan sivun muutosten päivittämistä tietokantaan.



Kuva 12. Yksittäisen kurssisivun näkymä.

Ylläpitäjä pääsee siis kolmella hiiren painalluksella mistä tahansa päin Voimanainen-sivustoa muokkaamaan esimerkiksi juuri kurssiviikon sivuja, eikä se vaadi muuta kuin sisäänkirjautumisen.

## Muokattavuuden haasteet

Edellä esitettiin, kuinka vaivattomasti muokkaaminen ja ylläpito tapahtuu. Esimerkkinä käytetyssä kurssiviikon sivujen muokkaamisessa näytettiin visuaalista näkymää, jonka avulla onkin usein mutkatonta muokata sisältöä halutulla tavalla. Haasteelliseksi sisällön muokkaamisen ja lisäämisen voi kuitenkin ylläpitäjälle tehdä kurssin toteuttamiseen käytetty ratkaisumalli esimerkiksi kurssin tehtävien suhteen, sillä niiden aikaansaami-

seksi jouduttiin lisäämään HTML-kieltä itse kurssiviikon sisältöön. Tästä on esimerkki esimerkikoodissa 2. Esimerkin HTML-ohjelmointikieltä sisältävässä lyhyessä otteessa tulostetaan ja muotoillaan ensimmäisen kurssiviikon ensimmäinen tehtäväosa. Linkki kurssiviikon tehtäviin luodaan p-elementti, jolle annetaan luokiksi *task-link*, *text-with-icon* ja *underline*. Näitä luokkia käytetään niiden sisällä olevan tekstin muotoiluun, asettamaan tehtävälinkeissä käytettävä kuva linkin eteen, ja siksi, että juuri nämä elementit voitaisiin poimia muiden p-elementtien joukosta JavaScriptillä.

Kun p-elementti on suljettu, luodaan uusi section-elementti, jolle annetaan luokka *task-section*. Tätä section-elementtiä käytetään tehtävien säiliönä, joka on oletusarvoisesti piilotettuna, mutta joka kuitenkin voidaan avata klikkaamalla edellä olevan p-elementin sisältötekstiä. Lisäksi tehtävissä esitetyn kysymyksen jälkeen luotiin tyhjä textarea-elementti, jonka tarkoitus on toimia vastausalustana käyttäjälle. Lopuksi vielä section-elementti suljetaan, kun kaikki tehtäväosion kysymykset on listattu sen sisään.

```
<p class="task-link text-with-icon underline">Tehtävä:
Kuka olet Voimanainen?</p>

<section class="task-section"><strong>Kuka olet Voima-
nainen</strong>

Kerro kuka olet nyt, aloittaessasi tämän kurssin, tä-
män matkan?

<textarea></textarea>

</section>
```

Esimerkkikoodi 2. HTML-ohjelmointikieltä tehtäväosiosta.

Mikäli asiakas jostain syystä haluaa muokata esimerkiksi juuri tehtäväosiota, hän joutuu tämänkaltaisen haasteen eteen. Tästä syystä onkin tärkeää, että asiakkaalle toimitetaan tarkat ohjeet sivuston hallintaa varten.

## 5 Sivuston dynaamisuus ja testauksen aloitus

Insinööriyönä rakennettua kurssisivustoa oli myös toissijaisesti tarkoitus suunnitella siten, että sitä pystyttäisiin käyttämään muihinkin vastaaviin käyttötarkoituksiin. Tämä huomioitiin muun muassa eri funktioiden suunnittelussa ja niiden käytössä siten, että sisältöä haetaan tunnistenumeron perusteella eikä esimerkiksi nimen perusteella. Tämä mahdollistaa sen, että sivustoon suunniteltua teemaa voidaan helposti käyttää

muissakin projekteissa osittain samalla tietokannalla, koska sivuista voidaan muuttaa sisältö helposti, eikä muutos riko sisällön linkitystä.

Mikäli sivustoa halutaan käyttää esimerkiksi toisen vastaavanlaisen kurssin luomiseen, tarvitsee tehdä lähinnä pieniä rakenteellisia muutoksia siihen, missä ja miten sisältöä halutaan esittää. Julkipuolen ulkoasun muokkaamisen pitäisi hoitua kätevästi tyylitiedostoon tehtävillä muutoksilla ja kuvatiedostojen päivittämisellä.

Kurssipuolen ulkoasu on valmiiksi melko neutraali: se käyttää pääasiassa kolmea väriä, joita muokkaamalla teemaa on helppo muokata haluttuun suuntaan. Lisäksi suurin osa kurssilla näkyvistä kuvista on sidottu vain sisältöön, joten niiden muokkaaminen onnistuu suoraan WordPressin omasta hallintapaneelistä samalla tavalla, kuin kurssien sisältöä muutenkin muokattaisiin. Muut kuvamuutokset onnistuvat esimerkiksi päivittämällä uudet käytettävät kuvat samannimisiksi, kuin vanhat olivat. Tässä tavassa mitään muutoksia itse koodiin ei välttämättä tarvitse tehdä.

Kuten luvussa 3.1 mainittiin, sivuston teema käyttää alatunnisteessa hyväkseen WordPressin omaa *wp\_nav\_menu* -funktioita ja dynaamista sivupalkkia, joka kutsuu WordPressin hallintapaneelissa sille osoitettuja pienoishjelmia. Tämän vuoksi julkiosan alatunniste on täysin dynaaminen, eli sen sisältö ja sisältötyyppi määritellään hallintapaneelistä ja sitä voidaan siten muokata sivuston käyttötarkoituksen mukaisesti millä palvelimella tahansa.

Insinööritöössä rakennetun sivuston testaus aloitettiin 9.4.2013. Testauksen alkua viivästivät ongelmat, jotka johtuivat ilmeisesti Metropolian palvelimen käyttöäoikeuksista. Ongelmat olivat jo entuudestaan tunnettuja WordPressin kanssa, ja sivusto siirrettiin toiselle palvelimelle, jotta testaus pääsi alkamaan. Testikäyttäjät olivat pääosin insinööritöön asiakkaan, Marja-Leena Savimäen, itse pyytämiä ja valitsema henkilöitä. Testaus tehtiin hyvin yksinkertaisesti jakamalla verkko-osoite sivuston testausversioon ja testikäyttäjän käyttäjätunnukset, joilla testikäyttäjä pääsi kirjautumaan sivulle ja kokeilemaan kurssisivuston puolta.

Testikäyttäjiltä saatiin palautetta melkoisen pian, ja palautteesta ilmeni, että sivuston kanssa oli tehty hyvää työtä, mutta puutteita ja epäselvyyksiäkin havaittiin. Etenkin fonttia, kirjainkokoja ja väriä keuhuttiin ja sivustoa kuvailtiin selkeäksi. Lisäksi tehtävien sisään rakentamista itse kurssin tekstin joukkoon kuvailtiin kivaksi ratkaisuksi, mutta

tehtäviin käsiksi pääsemisen epäiltiin mahdollisesti aiheuttavan ongelmia kokemattomille käyttäjille. Palautteessa pyydettiin myös lisäämään kurssiohjeet kurssisivun yläpalkin navigaatioon ja parantamaan itse ohjeiden sisältöä.

Testikäyttäjien palaute oli tärkeää ja selkeyttävää, sillä pääosin palautteessa heikon arvosanan saaneet osat olivat jo valmiiksi sellaisia, joita oli joko alettu kehittää paremmiksi tai joiden lisäkehitystä oli jo suunniteltu etukäteen. Palaute osoitti melko selvästi, että sivuston kehittäminen oli ollut ainakin jossain määrin onnistunutta ja että lisäkehittämisen aloittaminen ei ollut tarpeetonta.

## 6 Yhteenveto ja tulevaisuus

Insinööritöinä tehtiin vielä tämän raportin kirjoitusvaiheessa keskeneräinen, mutta lähitulevaisuudessa valmistuva verkkokurssituotteen alustana käytettävä verkkosivu. Valmiina sivusto tarjoaa vastinetta Voimanainen-kurssin kysynnälle. Jos kurssisivu onnistuu kysynnän tyydyttämisessä ja tyytyväisten käyttäjäkokemusten tuottamisessa, se vapauttaa myös insinööritöiden asiakkaan aikaa.

Insinööritöissä onnistuttiin käyttämään WordPressiä alustana kohtalaisen tehokkaasti, joskin on myönnettävä, että huippuosaaja olisi todennäköisesti siitä enemmänkin irti saanut. Tähän vaiheeseen kehittyntä lopputulosta voidaan kuitenkin pitää onnistuneena, etenkin kun otetaan huomioon, että verkkokurssisivuston rakentaminen on tähän asti yli puoli vuotta kestänyt katkonainen prosessi, jossa kuva onnistuneesta lopputuloksesta on muokkautunut matkan varrella useasti.

Varsinaisia isoja epäonnistumisia lopputyössä ei koettu, mutta pienehköjä epäonnistumisia sitäkin enemmän. Esimerkiksi alun perin helposti ratkaistavaksi kuviteltu PDF:n muodostaminen aiheutti harmia, vei paljon aikaa ja on ainakin tässä vaiheessa jäämässä toteutuksen ulkopuolelle. Lisäksi lopputyön aikana tehtiin ohjelmointiratkaisuja, jotka jälkikäteen ajateltuna tuntuvat vähintäänkin heikoilta, mutta joiden muuttaminen jälkikäteen olisi vaatinut liian suurta ajankäyttöä niistä saavutettavaan hyötyyn nähden. Tästä on esimerkkinä tunnisteiden käyttäminen liian monen HTML-elementin kanssa sen sijaan, että olisi hyödynnetty paremmin luokkia, jotka ovat kevyempiä CSS:n kannalta. Toisaalta insinööritöiden aikana tehdyt virheet olivat opettavaisia, ja niitä tuskin toistetaan.

Koska insinööri työ ei ole vielä valmis julkaistavaksi, jää tulevaisuuteen vielä paljon tehtävää. Verkkosivulta puuttuvat muun muassa verkkokauppaominaisuudet, mutta niiden suunnittelu on jo pitkällä. Verkkokauppana tullaan käyttämään suomalaista Checkout-maksupalvelua, joka osoittautui vertailussa edullisimmaksi ratkaisuksi ja josta kuultiin vain myönteistä palautetta. Checkoutin valintaa helpottaa myös se, että se tukee WordPressin eCommerce-liitännäistä, jota voidaan mahdollisesti hyödyntää myös verkkosivuilla.

Lisäksi tulevaisuuteen jää myös kysymys verkkokurssin käyttäjien vastausten tallentamisesta tietokantaan. Jatkossa tulee hankkia lisää palautetta testauksesta ja pohtia sitä kautta tallennuksen tärkeyttä. Tietojen tallentamisen teknistä ratkaisumallia on jo alettu pohtia, mutta parasta ratkaisumallia ei todennäköisesti ole vielä löydetty.

## Lähteet

- 1 About Drupal. Verkkodokumentti. Drupal. <<http://drupal.org/about>>. Luettu 23.4.2013.
- 2 About Wordpress. Verkkodokumentti. Wordpress. <<http://wordpress.org/about/>>. Luettu 14.4.2013.
- 3 Hall of Fame CMS: WordPress. Verkkodokumentti. Packt Publishing. <<https://www.packtpub.com/open-source-awards-home/2010-wordpress>> 14.4.2013.
- 4 Imel Ryan. 2011. WordPress wins 2011 Open Source Web App of the Year at The Critters. Verkkodokumentti. WPCandy. <<http://wpcandy.com/reports/wordpress-wins-2011-open-source-web-app-of-the-year-at-the-critters/>>. Luettu 14.4.2013.
- 5 Featured Themes. Verkkodokumentti. Wordpress. <<http://wordpress.org/extend/themes/>>. Luettu 23.4.2013.
- 6 The Loop. Verkkodokumentti. Wordpress. <[http://codex.wordpress.org/The\\_Loop](http://codex.wordpress.org/The_Loop)>. Luettu 29.4.2013.
- 7 Class PasswordHash. Verkkodokumentti. Wordpress. <<http://phpdoc.wordpress.org/trunk/phpass/PasswordHash.html>>. Luettu 23.4.2013.
- 8 Krug, Steve. 2006. Don't Make Me Think! A Common Sense Approach to Web Usability. Second Edition. USA: New Riders.
- 9 The art of Zen education resource: History of Zen Buddhism. Verkkodokumentti. National Gallery of Victoria. <[http://www.ngv.vic.gov.au/artofzen/ed\\_zen\\_history.html](http://www.ngv.vic.gov.au/artofzen/ed_zen_history.html)>. Luettu 23.4.2013.
- 10 Theme Development. Verkkodokumentti. Wordpress. <[http://codex.wordpress.org/Theme\\_Development](http://codex.wordpress.org/Theme_Development)>. Luettu 24.4.2013.
- 11 Custom Fields. Verkkodokumentti. Wordpress. <[http://codex.wordpress.org/Custom\\_Fields](http://codex.wordpress.org/Custom_Fields)>. Luettu 29.4.2013.
- 12 Function Reference/dynamic sidebar. Verkkodokumentti. Wordpress. <[http://codex.wordpress.org/Function\\_Reference/dynamic\\_sidebar](http://codex.wordpress.org/Function_Reference/dynamic_sidebar)>. Luettu 29.4.2013.

- 13 Function Reference/wp nav menu. Verkkodokumentti. Wordpress.  
<[http://codex.wordpress.org/Function\\_Reference/wp\\_nav\\_menu](http://codex.wordpress.org/Function_Reference/wp_nav_menu)>. Luettu 25.4.2013.
- 14 JavaScript Introduction. Verkkodokumentti. W3Schools.  
<[http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp)>. Luettu 30.4.2013.
- 15 .change(). Verkkodokumentti. jQuery Foundation.  
<<http://api.jquery.com/change/>>. Luettu 29.4.2013.
- 16 CSS3 Previews. Verkkodokumentti. CSS3.info. <<http://www.css3.info/preview/>>. Luettu 25.4.2013.
- 17 Chapman, Cameron. 2009. Simplicity in Good Web Design : Advantages & How - to. Verkkodokumentti. Noupe. <<http://www.noupe.com/design/simplicity-in-good-web-design-advantages-how-to.html>>. Luettu 30.4.2013.
- 18 Roles and Capabilities. Verkkodokumentti. Wordpress.  
<[http://codex.wordpress.org/Roles\\_and\\_Capabilities](http://codex.wordpress.org/Roles_and_Capabilities)>. Luettu 25.4.2013.
- 19 Custom Post Types. Verkkodokumentti. Wordpress.  
<[http://codex.wordpress.org/Post\\_Types#Custom\\_Post\\_Types](http://codex.wordpress.org/Post_Types#Custom_Post_Types)>. Luettu 25.4.2013.
- 20 Styling Page-Links. Verkkodokumentti. Wordpress.  
<[http://codex.wordpress.org/Styling\\_Page-Links](http://codex.wordpress.org/Styling_Page-Links)>. Luettu 25.4.2013.
- 21 TinyMCE. Verkkodokumentti. Wordpress.  
<<http://codex.wordpress.org/TinyMCE>>. Luettu 28.4.2013.
- 22 Who is using TinyMCE? Verkkodokumentti. Moxiecode Systems AB.  
<<http://www.tinymce.com/enterprise/using.php>>. Luettu 28.4.2013.

